NASA Technical Memorandum 4698

Performance Trend of Different Algorithms for Structural Design Optimization

Surya N. Patnaik, Rula M. Coroneos, James D. Guptill,
and Dale A. Hopkins

OCTOBER 1996

National Aeronautics and
Space Administration

NASA Technical Memorandum 4698

Performance Trend of Different Algorithms for Structural Design Optimization

Surya N. Patnaik
Ohio Aerospace Institute
Brook Park, Ohio

Rula M. Coroneos, James D. Guptill, and Dale A. Hopkins
Lewis Research Center
Cleveland, Ohio

National Aeronautics and
Space Administration

Office of Management

Scientific and Technical
Information Program

1996

Performance Trend of Different Algorithms for Structural Design Optimization

Surya N. Patnaik
Ohio Aerospace Institute
Brook Park, Ohio 44142

Rula M. Coroneos, James D. Guptill, and Dale A. Hopkins
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Summary

Nonlinear programming algorithms play an important role in structural design optimization. Fortunately, several algorithms with computer codes are available. At NASA Lewis Research Center, a project was initiated to assess performance of different optimizers through the development of a computer code CometBoards. This paper summarizes the conclusions of that research. CometBoards was employed to solve sets of small, medium and large structural problems, using different optimizers on a Cray-YMP8E/8128 computer. The reliability and efficiency of the optimizers were determined from the performance of these problems. For small problems, the performance of most of the optimizers could be considered adequate. For large problems however, three optimizers (two sequential quadratic programming routines, DNCONG of IMSL and SQP of IDESIGN, along with the sequential unconstrained minimizations technique SUMT) outperformed others. At optimum, most optimizers captured an identical number of active displacement and frequency constraints but the number of active stress constraints differed among the optimizers. This discrepancy can be attributed to singularity conditions in the optimization and the alleviation of this discrepancy can improve the efficiency of optimizers.

Introduction

Nonlinear programming algorithms play an important role in structural design optimization. Fortunately, several algorithms with computer codes have been developed during the past few decades. To assess performance of different optimizers, a project was initiated at NASA Lewis Research Center and a computer code called CometBoards, which is an acronym for Comparative Evaluation Test Bed of Optimization and Analysis Routines for the Design of Structures (ref. 1), was developed.

Because licenses for some of the optimization codes has expired, numerical results are provided herein for six optimizers. CometBoards incorporates about a dozen popular optimization codes. These are: the feasible directions method (FD; ref. 2); fully utilized design (FUD; ref. 3); genetic algorithm (GENMO; ref. 4); generalized reduced gradient method (GRG; ref. 5); the DNCONG of the IMSL routine (ref. 6); modified feasible direction method (MFD; ref. 7); NPSOL, which is available in the NAG library (ref. 8); the optimality criteria methods (OC; ref. 3); the reduced gradient method (RG; ref. 9); the sequential linear programming method (SLP; ref. 2); the sequential quadratic programming technique (SQP of IDESIGN; ref. 10); the sequence of unconstrained minimizations technique (SUMT; ref. 11); and the cascade strategy, which includes more than one optimization algorithm (ref. 12). CometBoards was employed to solve a set of 41 structural problems by using its eight optimizers on a Cray-YMP8E/8128 computer. The reliability and efficiency of the eight optimizers were ascertained on the basis of the performance of these problems. The problems were solved for multiple load conditions, and behavior constraints were imposed on stresses, displacements, and frequencies. The examples were selected so that at optimum, numerous stress, displacement, and frequency constraints were active. Initial design, upper and lower bounds, and convergence parameters were specified to ensure that the evaluation had no bias towards any particular optimizer or any particular problem. The eight optimizers might have been updated during the time CometBoards was developed, but any such improvements were not accounted for.

Evaluations of optimizers that are available in the literature (refs. 13 to 21) deal broadly with individual code validation by their developers. The studies lacked uniformity because problems and computational platforms differed and the evaluations were over a decade old. For example, Arora, (refs. 14 and 15), the developer of SQP of IDESIGN, compared his algorithm to the NAG/NPSOL optimizer. Most of Arora's problems were trusses for stress and displacement constraints and were solved on a PRIME 750 computer. Schittkowski, who is the developer of the DNCONG optimization routine in the IMSL library, essentially validated his code (refs. 16 and 17) by solving many theoretical examples on a Telefunken-TR-400 computer. Venkayya (ref. 18), one of the developers of ASTROS, in which OC and FD optimizers are used (ref. 19), attempted an evaluation of a few practical problems on a VAX 11/785 computer. An intermediate complexity wing problem, used by Venkayya with stress and displacement constraints (ref. 18), is also included in our test bed with the addition of frequency constraints. Ragsdell's evaluation (refs. 20 and 21) includes mostly simple mechanical application problems. Our current paper differs from those available in the literature in several respects: (1) a single tool, CometBoards, evaluates all eight optimizers on a common Cray-YMP computer; (2) solutions to a set of problems, which were grouped into categories of small, medium, and large, are used; and (3) design parameters were selected to ensure that the evaluation had no bias towards problems or optimizers. In brief, the comprehensive evaluation presented in this paper does not duplicate previous work. This paper presents a brief theory of optimization methods, a description of CometBoards, a summary of the numerical examples and their solutions, discussion, and conclusions.

Symbols

FD	feasible directions
FUD	fully utilized design
GENMO	genetic algorithm
GRG	generalized reduced gradient method
IMSL	international mathematical subroutine library
MFD	modified feasible direction method
NPSOL	nonlinear programming package of the systems optimization laboratory
OC	optimality criteria
RG	reduced gradient
SLP	sequential linear programming
SQP	sequential quadratic programming
SUMT	sequence of unconstrained minimizations technique

Theory of Optimization Methods

Structural design can be formulated as: Find the n design variables $\vec{x}$, within prescribed upper and lower bounds ($x_i^L \leq x_i \leq x_i^U$, $i = 1,2,\ldots,n$) which make a scalar objective function $f(\vec{x})$ an extremum (here, minimum weight) subject to: a set of m_i inequality constraints $g_j(\vec{x}) \geq 0$, ($j = 1,2,\ldots,m_i$) and m_e equality constraints $g_{j+m_i}(\vec{x}) = 0$ ($j = 1,2,\ldots,m_e$).

Stress, displacement, and frequency behavior constraints were investigated in this study. A cursory account of representative optimization methods available in CometBoards is provided herein. Readers may refer to specified references for details.

(1) The sequence of unconstrained minimizations technique (SUMT), as implemented in the code NEWSUMT, is available in CometBoards. In NEWSUMT, the penalty function has been modified to improve efficiency and a modified Newton's approach is used to calculate the direction vector while a golden section technique is used to determine step length.

(2) Sequential linear programming (SLP), as implemented in design optimization tools (DOT 2.0) is available in CometBoards. From the original nonlinear problem, a linear programming subproblem is obtained by linearizing a set of critical constraints and the objective function around a design point. The linearization process and linear solution sequence is repeated until convergence is achieved.

(3) The method of feasible directions (FD), as implemented in DOT 2.0, is available in CometBoards. In FD, a usable feasible direction is used. A minimum along the search direction is generated by polynomial approximation.

(4) SQP of IDESIGN, DNCONG of IMSL, and NPSOL in NAG, three implementations of the sequential quadratic programming technique, are available in CometBoards. In this technique, the original nonlinear problem is solved through a sequence of quadratic subproblems. In SQP of IDESIGN, a Lagrangian function is approximated. The step length is obtained by minimizing a composite descent function. DNCONG of IMSL uses quasi-Newton updates for the Hessian of the Lagrangian function while the constraints are linearized (ref. 22). The step length for an augmented Lagrangian is calculated using a bisection method (ref. 23). NPSOL in NAG also uses an augmented Lagrangian. The search direction is generated through a quadratic subproblem while step length is calculated using an augmented Lagrangian, which is designed to avoid discontinuities as much as possible.

(5) The reduced gradient method (RG), as implemented in the code OPT, has been incorporated into CometBoards. This method partitions the design variable into decision and slave variables and a reduced gradient is used to generate a search direction. A line search is carried out by bounding the minimum and then calculating the minimum within some tolerance.

(6) The optimality criteria method (OC), available in CometBoards, can be considered as a variant of the Lagrange multiplier approach applied to structural design problems. In OC, an iterative scheme is followed to update the multipliers and the design variables separately.

Description of CometBoards

The basic organization of CometBoards is depicted in figure 1. The central executive with command level interface (fig. 1) links the three modules (optimizer, analyzer, and data input) of the code to formulate and solve an optimization problem. The analyzer options are the displacement method (refs. 8 and 20),

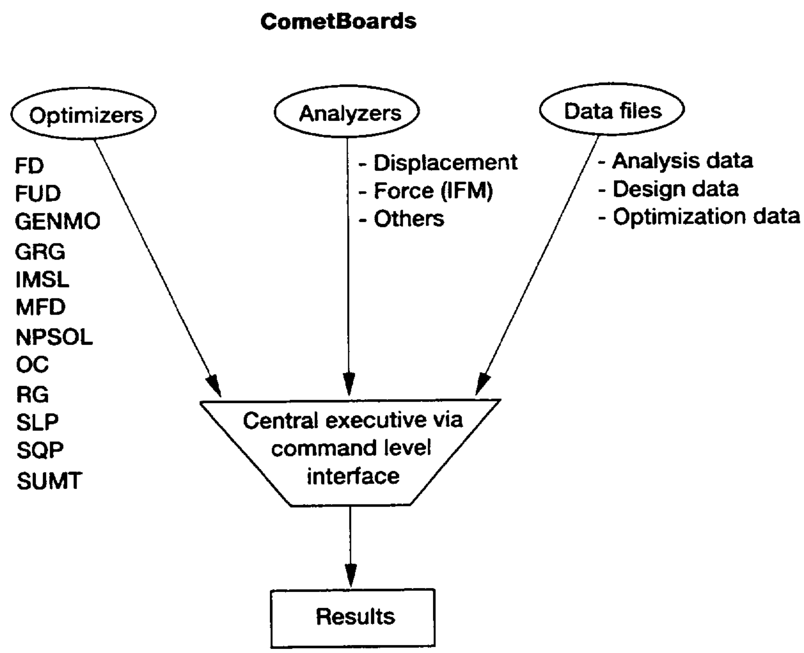

Figure 1.—Comparative evaluation test bed of optimization and analysis routines for the design of structures (CometBoards).

the integrated force method (refs. 3 and 25), and the simplified force method (ref. 3), etc. There are three input data files, one for analysis (anldat), one for design (dsgndat), and one for optimization (optdat). CometBoards has considerable flexibility in solving a design problem by choosing any one of the available optimizers and any one of three analyzers. A more detailed description of CometBoards can be found in the reference 1, User's Manual.

Example Problems

The numerical test bed of CometBoards includes over 41 problems, most of which were taken from the literature (refs. 1, 3, 18, and 26 to 31). Minimum weight was the objective and a linking strategy was followed to reduce the number of design variables. Stress, displacement, and frequency behavior constraints were considered. Multiple static load conditions and consistent elemental mass for dynamic analyses were also considered. The load conditions, mass distributions, and behavior limitations were specified to ensure that several types of behavior constraints were active at the optimum. The initial design of unity was considered for all problems unless otherwise specified. Each problem had a consistent set of upper and lower bounds specified. Typically, default optimization parameters and convergence criteria specified in the individual codes were used. These parameters, however, were changed when convergence difficulty was encountered. Results for all 41 examples are summarized in table I. The normalized optimum weight and the normalized Cray-YMP8E/8128 CPU time for a select set of 14 examples are given in table II and depicted in figures 2 to 5. The weight was normalized with respect to the optimum weight obtained for the best feasible design. A brief description of the 14 examples follows.

Examples P1a to P1d: 3-Bar Truss

The popular 3-bar truss (refs. 3, 26, 30), as shown in figure 6, (with modulus E = 30 000 ksi, and density $\rho = 0.1$ lb/in.^{3}) was subjected to a single load condition. It had three design variables, and six constraints (3 stress, two displacement and one frequency). Optimum weight and CPU time are depicted in table II (P1a, P1b, P1c, P1d), and figures 2 and 5. The optimum weight was 92.87 lb and one stress, one displacement, and one frequency constraints were active. Five optimizers (SUMT, SQP, IMSL, NPSOL and RG) performed satisfactorily. OC was inadequate, yielding a 38.6 percent over-design. The problem was solved again for three different initial designs (the SUMT optimum design, 150 percent of SUMT optimum, and 50 percent of SUMT optimum). Results followed the pattern of the earlier problem where the initial design was unity. The CPU times on the Cray-YMP computer required for different optimizers are depicted in figure 5. For unit initial design, SQP required the least CPU time of 0.14 sec, while RG was most expensive at 3.18 sec.

Example P2: Tapered 10-Bar Truss

A tapered 10-bar aluminum truss (ref. 3), shown in figure 7, was subjected to two load conditions. It had 10 design variables,

TABLE I.— SUMMARY FOR 41 TEST BED PROBLEMS

Problem number	Problem description and number of design variables	Constraints specified	Active constraints for optimization codes					
			SUMT	SQP	IMSL	NPSOL	RG	OC
P1a	3-bar truss (3 IDV, ID = 1)	3S, 2D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	[a]1F
P1b	3-bar truss (3 IDV, ID = OPT)	3S, 2D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	[a]1F
P1c	3-bar truss (3 IDV, ID = 1.5 × OPT)	3S, 2D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	[a]1F
P1d	3-bar truss (3 IDV, ID = 0.5 × OPT)	3S, 2D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	1S, 1D, 1F	[a]1F
P2	Tapered 10-bar truss (10 IDV)	20S, 4D, 1F	7S, 2D, 1F	8S, 2D, 1F	8S, 2D, 1F	8S, 2D, 1F	[a]1S, 1F	[a]5S, 2D, 1F
P3	Tapered cantilever beam (8 IDV)	16S, 4D, 1F	6S	6S	6S	6S	6S	4S
P4	25-bar truss (8 LDV)	50S, 36D	4D	4D	4D	4D	[a]2D	(a)
P5	165 feet tall antenna tower (6 LDV)	504S, 24D, 1F	[a]7S, 12D, 1F	6S, 12D, 1F	6S, 12D, 1F	6S, 12D, 1F	[a]1F	(a)
P6	60-bar trussed ring (25 LDV)	180S, 3D, 1F	21S, 1D, 1F	21S, 1D, 1F	21S, 1D, 1F	19S, 1D, 1F	[a]18S, 1D, 1F	[a]10S, 1F
P7	Geodesic dome (12 LDV)	252S, 1D, 1F	162S, 1D, 1F	168S, 1D, 1F	156S, 1D, 1F	162S, 1D, 1F	[a]18S, 1D	[a]12S
P8a	Intermediate Complexity Wing (57 LDV, ID = 1.0)	316S, 4D, 1F	106S, 1D	117S, 1D, 1F	117S, 1D, 1F	75S, 1D, 1F	[a]6S	[a]19S
P8b	Intermediate Complexity Wing (57 LDV, ID = OPT)	316S, 4D, 1F	(a)	106S, 1D	117S, 1D, 1F	117S, 1D, 1F	106S, 1D	[a]42S
P8c	Intermediate Complexity Wing (57 LDV, ID = 1.5 × OPT)	316S, 4D, 1F	109S, 1D	[a]115S, 1D, 1F	[a]88S, 1D, 1F	[a]2S	[a]7S	(a)
P8d	Intermediate Complexity Wing (57 LDV, ID = 0.5 × OPT)	316S, 4D, 1F	117S, 1D, 1F	[a]99S, 1D, 1F	(a)	[a]14S	[a]7S	(a)
P9	10-bar truss (10 IDV)	20S, 4D, 1F	7S, 2D, 1F	8S, 2D, 1F	8S, 2D, 1F	8S, 2D, 1F	[a]1F	[a]5S, 2D, 1F
P10	10-bar truss (5 LDV)	20S, 4D, 1F	6S, 2D, 1F	6S, 2D, 1F	6S, 2D, 1F	6S, 2D, 1F	[a]3S, 1F	6S, 2D, 1F
P11	Stiffened 10-bar truss (18 IDV)	36S, 4D, 1F	8S	9S	8S	9S	[a]1S	5S
P12	Stiffened 10-bar truss (2 LDV)	36S, 4D, 1F	2S	2S	2S	2S	[a]1F	2S
P13	Cantilever membrane (8 IDV)	16S, 4D, 1F	6S	6S	6S	6S	6S	4S
P14	Cantilever membrane (1 LDV)	16S, 4D, 1F	1S	1S	1S	1S	1S	1S
P15	Cantilever membrane 16-quad elements (16 IDV)	32S, 4D, 1F	1S, 4D	(a)	4D	[a]4S	[a]5S	4D
P16	Cantilever membrane 32-quad elements (16 LDV)	64S, 4D, 1F	1S, 4D	4D	4D	[a]9S	[a]4S	4D
P17	Cantilever membrane 48-quad elements (16 LDV)	96S, 4D, 1F	13S, 4D	[a]13S, 4D	15S, 4D	[a]8S	[a]3S	13S, 2D
P18	Cantilever membrane 64-quad elements (16 LDV)	128S, 4D, 1F	32S, 4D	[a]29S, 4D	31S, 4D	[a]16S	(a)	23S, 2D
P19	60-bar trussed ring (25 LDV)	180S	38S	[a]35S	38S	38S	[a]14S	40S
P20	60-bar trussed ring (25 LDV)	3D	1D	1D	1D	1D	[a]1D	1D
P21	60-bar trussed ring (25 LDV)	1F	(a)	1F	1F	1F	[a]1F	[a]1F
P22	60-bar trussed ring (25 LDV)	180S, 24D	28S, 1D	[a]30S, 1D	27S, 1D	29S, 1D	[a]20S	18S, 1D
P23	60-bar trussed ring (25 LDV)	24D, 1F	[a]1D, 1F	1D, 1F	1D, 1F	[a]1D, 1F	[a]1F	[a]1D, 1F
P24	Stiffened 60-bar trussed ring (49 LDV)	252S	75S	75S	75S	75S	75S	[a]59S
P25	Stiffened 60-bar trussed ring (49 LDV)	3D	1D	[a]1D	1D	(a)	1D	1D
P26	Stiffened 60-bar trussed ring (49 LDV)	1F	1F	1F	1F	(a)	1F	1F
P27	Stiffened 60-bar trussed ring (49 LDV)	252S, 24D	75S, 1D	75S, 1D	75S, 1D	75S, 1D	76S, 1D	[a]17S
P28	Stiffened 60-bar trussed ring (49 LDV)	24D, 1F	1D, 1F	1D, 1F	1D, 1F	[a]1D	1D, 1F	1D, 1F
P29	Stiffened 60-bar trussed ring (49 LDV)	252S, 3D, 1F	44S, 1F	46S, 1F	46S, 1F	46S, 1F	47S, 1F	[a]3S
P30	Stiffened ring (24 IDV)	72S	28S	28S	28S	28S	27S	28S
P31	Stiffened ring (24 IDV)	3D	1D	1D	1D	(a)	1D	1D
P32	Stiffened ring (24 IDV)	1F	1F	1F	1F	(a)	[a]1F	1F
P33	Stiffened ring (24 IDV)	72S, 24D	28S, 1D	27S, 1D	28S, 1D	28S, 1D	25S, 1D	[a]18S, 1D
P34	Stiffened ring (24 IDV)	24D, 1F	1D, 1F	1D, 1F	1D, 1F	[a]2D	1D, 1F	1D, 1F
P35	Stiffened ring (24 IDV)	72S, 3D, 1F	17S, 1F	17S, 1F	17S, 1F	17S, 1F	17S, 1F	[a]16S, 1F

[a]Optimum weight obtained differs by more than 5 percent or constraint violation more than 1 percent (see ref. 1).
IDV: Independent design variable   ID: Initial design   LDV: Linked design variable   OPT: SUMT optimum design   D: Displacement constraints   F: Frequency constraints
S: Stress constraints

TABLE II.—OPTIMUM WEIGHT AND CRAY-YMP 8E/8128 CPU TIME FOR SELECTED SET OF EXAMPLE PROBLEMS

Problem number	Optimization methods											
	SUMT		SQP		IMSL		NPSOL		RG		OC	
	Weight[b]	CPU	Weight[b]	CPU	Weight[b]	CPU	Weight[b]	CPU	Weight[b]	CPU	Weight[b]	CPU
P1a	1.001	1.799	1.000	1.000	1.000	1.972	1.000	2.076	1.000	22.069	1.386	10.257
P1b	1.001	7.833	1.000	1.000	1.000	9.633	1.000	155.267	1.000	5.567	1.386	49.233
P1c	1.001	1.588	1.000	1.000	1.000	1.662	1.000	31.818	1.000	2.000	1.386	10.000
P1d	1.001	1.926	1.000	1.000	1.000	2.733	1.000	14.444	1.000	16.452	1.386	10.985
P2	1.000	1.500	1.000	1.000	1.000	1.133	1.000	1.022	(Failed)	---	1.056	9.324
P3	0.999	1.268	1.000	1.000	1.000	0.976	1.001	3.169	1.000	6.121	1.028	16.834
P4	1.000	2.533	1.000	1.000	1.000	1.539	1.000	3.759	(Failed)	---	(Failed)	---
P5	[a]0.940	1.065	1.019	1.000	1.019	1.134	1.017	5.026	(Failed)	---	2.773	4.62
P6	1.000	1.605	1.000	1.000	1.000	1.120	1.000	3.899	1.832	20.716	[a]1.041	8.279
P7	1.021	0.538	1.000	1.000	1.015	0.550	1.016	0.658	2.022	0.066	2.976	4.456
P8a	1.004	1.116	1.000	1.000	1.000	1.695	1.037	7.712	(Failed)	---	1.201	1.571
P8b	0.790	1.680	1.000	1.000	1.000	5.884	1.000	3.323	1.004	0.089	1.350	8.076
P8c	1.000	1.000	[a]0.998	0.571	[a]1.077	0.602	1.346	0.244	1.471	0.627	1.000	0.666
P8d	1.000	1.000	[a]0.981	0.244	(Failed)	---	[a]0.501	0.028	(Failed)	---	1.000	0.276

[a]Infeasible design.
[b]Normalized weight.

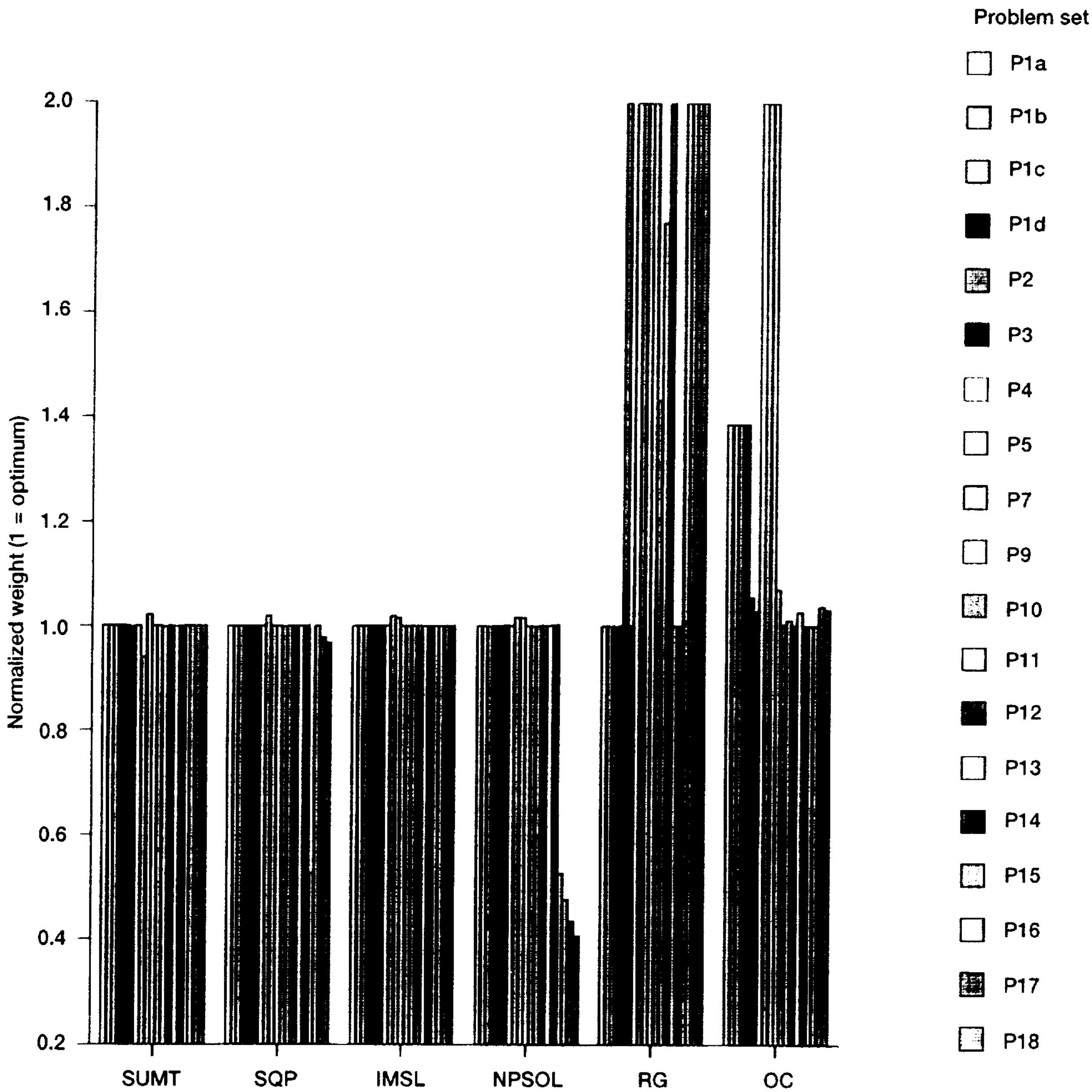

Figure 2.—Performance of different optimizers for small problems.

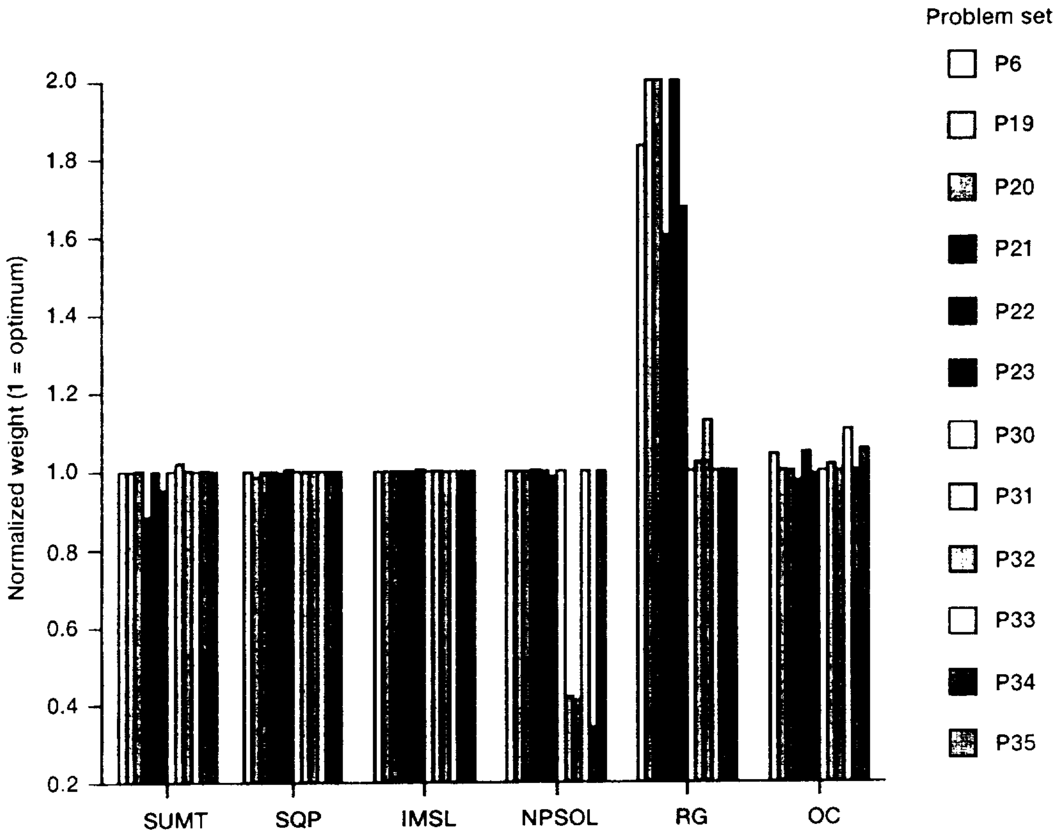

Figure 3.—Performance of different optimizers for medium problems.

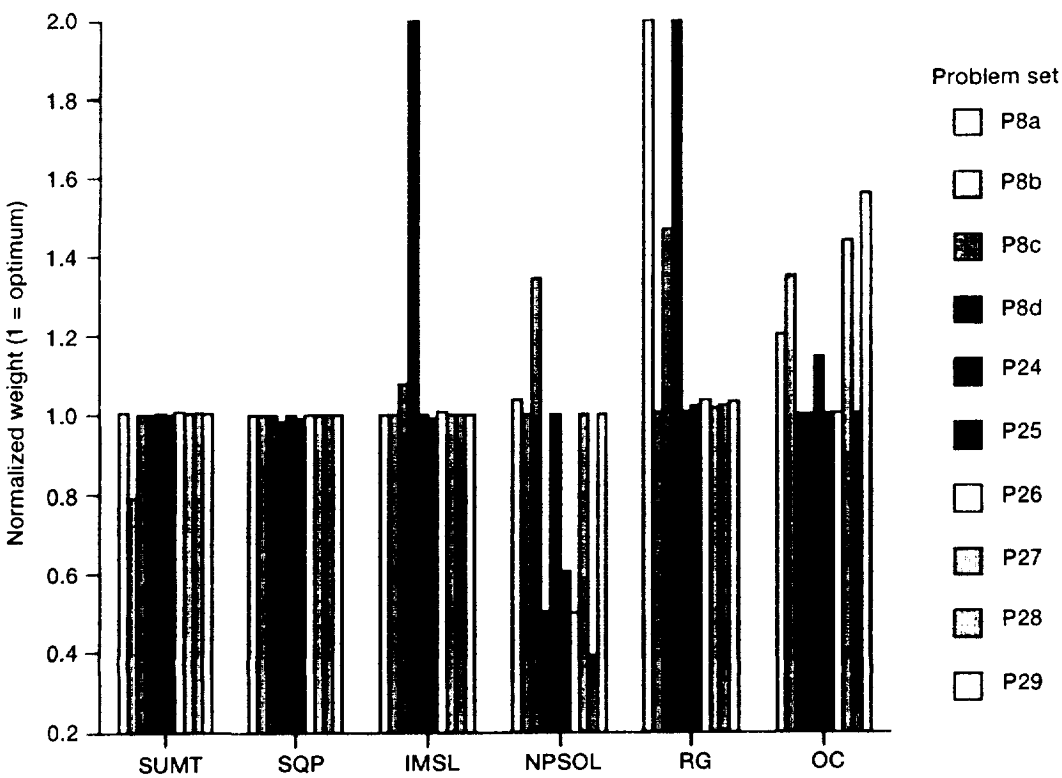

Figure 4.—Performance of different optimizers for large problems.

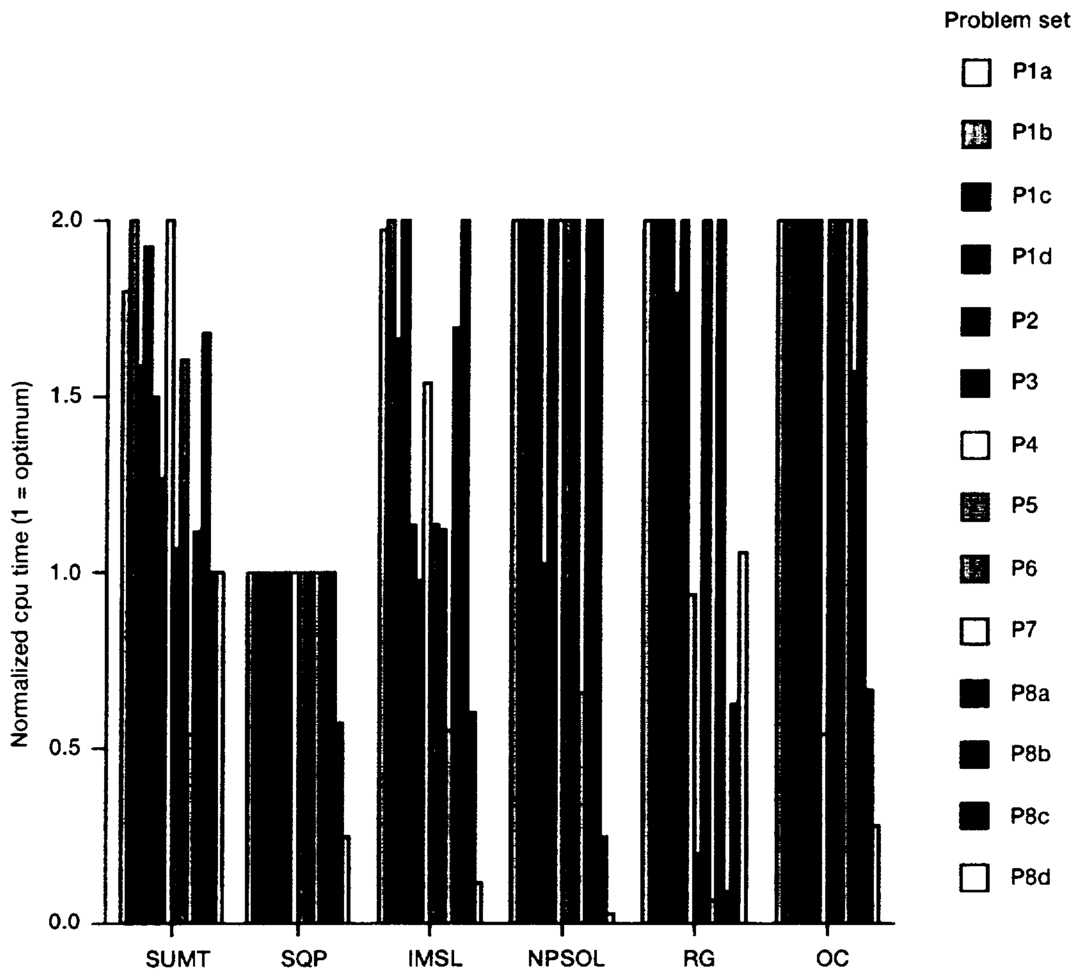

Figure 5.—Cray-YMP cpu time for different optimization methods for 14 problems.

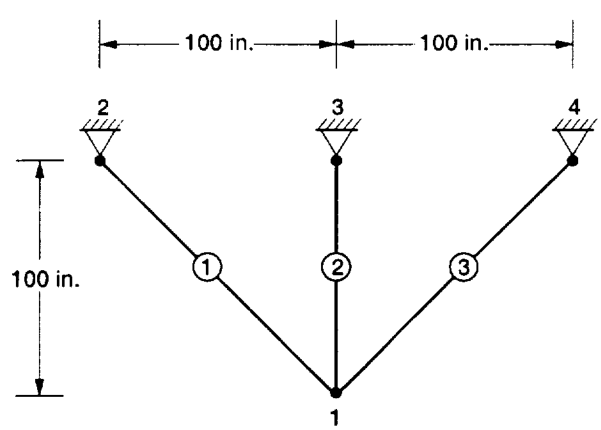

Figure 6.—Three-bar truss. (Elements are circled, nodes are not.)

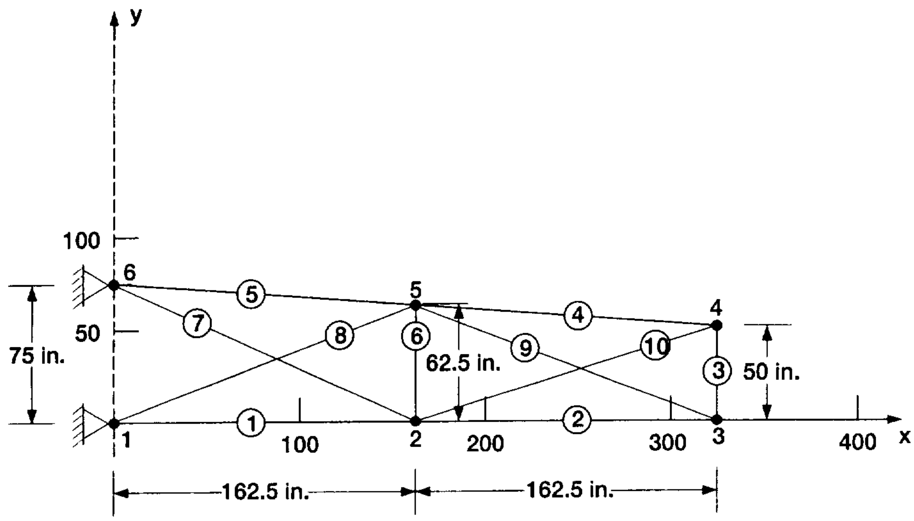

Figure 7.—Tapered ten-bar truss. (Elements are circled, nodes are not.)

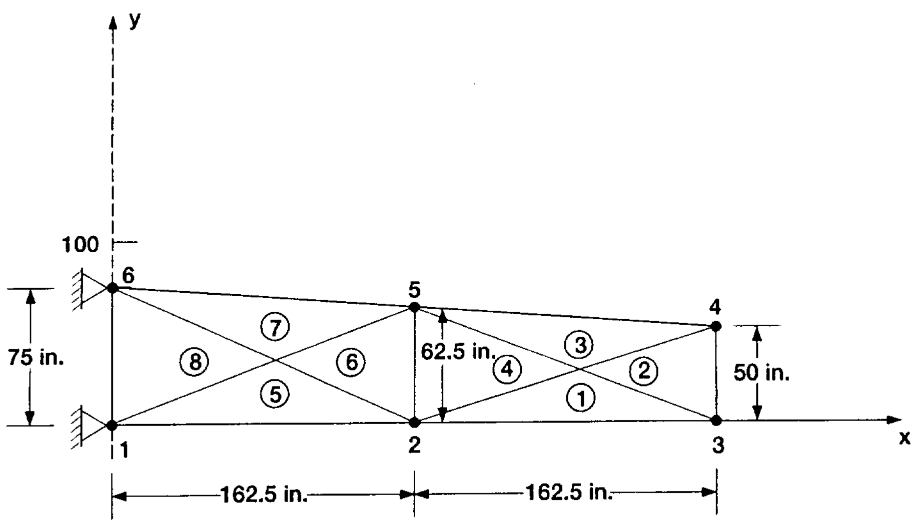

Figure 8.—Tapered cantilever beam modeled with eight triangular membrane elements. (Elements are circled, nodes are not.)

and 25 behavior constraints (20 stress, four displacement, and one frequency). The optimum weight was 3326.74 lb with 11 active constraints (8 stress, two displacement, and one frequency). Four optimizers (SUMT, SQP, IMSL, and NPSOL) converged for this example. Optimizer RG failed, and OC was marginal at 5.6 percent over-design. Cray-YMP CPU time varied between 1.28 sec for NPSOL and 1.91 sec for SUMT.

Example P3: Tapered Cantilever Beam

The cantilever truss of example P2, was modeled next using 8 triangular membrane elements, as shown in figure 8. The loads and constraints were identical to example P2. The eight thicknesses of the elements were considered the 8 design variables. The problem had 21 constraints (16 von Mises stress, four displacement and one frequency). Optimum results obtained are given in table II. The optimum weight was 1440.24 lb with six active stress constraints. Five optimizers (SUMT, SQP, IMSL, NPSOL, and RG) performed well while OC produced a 2.8-percent over-design (see table II). Cray-YMP CPU time varied from 1.79 sec for IMSL to 11.22 sec for RG.

Example P4: 25-Bar Truss

A 25-bar aluminum truss (refs. 26 and 27), as shown in figure 9, had 8 linked design variables, and was subjected to two load conditions. It had a total of 86 behavior constraints,

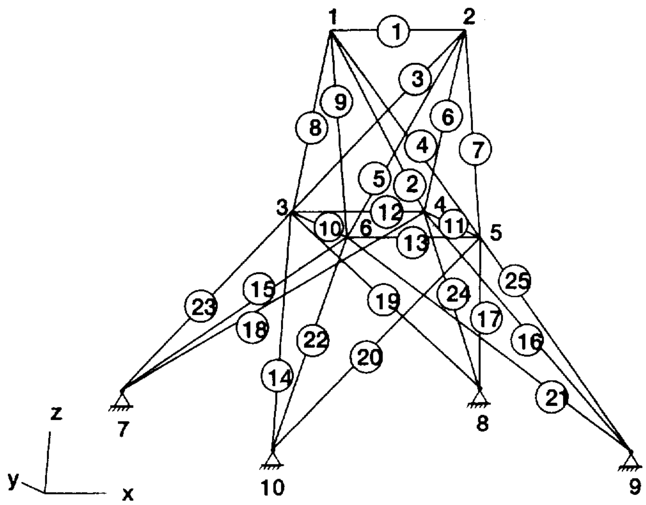

Figure 9.—Twenty-five bar-truss. (Elements are circled, nodes are not.)

(50 stress and 36 displacement). Four optimizers (SUMT, SQP, IMSL, and NPSOL) converged to an optimum weight of 544.73 lb with four active displacement constraints (tables I and II). Optimizers RG and OC failed. Cray-YMP CPU time ranged from 1.64 sec for SQP to 6.15 sec for NPSOL.

Example P5: 165-Ft-Tall Antenna Tower

A 165-ft-tall steel antenna tower with 252 members, as depicted in figure 10, (ref. 26), had six linked design variables and was subjected to two load conditions. Its overhead dish antenna was modeled as a lumped mass for frequency calculations. It had a total of 529 behavior constraints (504 stress, 24 displacement, and one frequency). Three optimizers (SQP, IMSL, and NPSOL) converged to an optimum solution of 5299.84 lb with small deviations (table II). At optimum, six stress, 12 displacement, and one frequency constraints were active. Optimizers RG and OC failed while SUMT produced a six percent under-design. The Cray-YMP CPU time varied between 376.83 sec for SQP and 1893.80 sec for NPSOL.

Example P6: 60-Bar Trussed Ring

A 60-bar trussed aluminum ring (ref. 3) was subjected to three load conditions and had two lumped masses, as depicted in figure 11. It had a total of 184 constraints (180 stress, three displacement, one frequency) and 25 linked design variables. The optimum weight was 414.51 lbs, and at optimum, 22 stress, one displacement, and one frequency constraints were active. Four optimizers (SUMT, SQP, IMSL, and NPSOL) converged (table II). Optimizer RG failed, whereas OC produced a 4.1 percent over-design with a 1.1 percent constraint violation. Cray-YMP CPU solution time ranged from 36.96 sec for SQP to 144.11 sec for NPSOL.

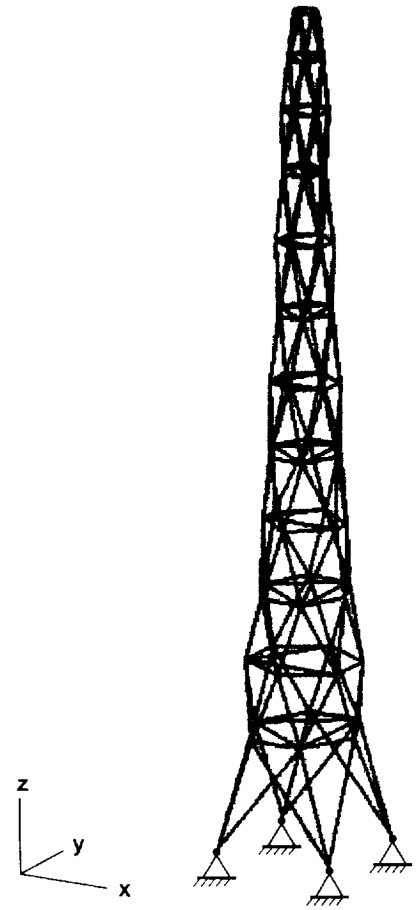

Figure 10.—One-hundred-sixty-five-ft tall antenna tower.

Example P7: Geodesic Dome

A geodesic dome (refs. 28 and 29), shown in figure 12 with a diameter of 240 in. and height of 30 in., was subjected to a single load condition. It was modeled using 156 bars and 96 triangular membrane elements. The bars were made of a material with modulus E = 30 000 ksi, and density ρ = 0.1 lb/in.^{3} Membranes were made of aluminum, with modulus E = 10 000 ksi, and density ρ = 0.1 lb/in.^{3} The bar areas and membrane thicknesses were grouped to obtain eight and four linked design variables, respectively. The dome had a total of 254 constraints, (156 stresses for bars, 96 von Mises stresses for membranes, one displacement, and one frequency). The optimum weight obtained was 1022.67 lb with 170 active constraints, (168 stress constraints, one displacement, and one frequency (table I). Four optimizers (SUMT, SQP, IMSL, and NPSOL) converged with small deviations. Optimizers RG and OC failed. The Cray-YMP CPU time varied between 448.32 sec for SUMT to 548.36 sec for NPSOL.

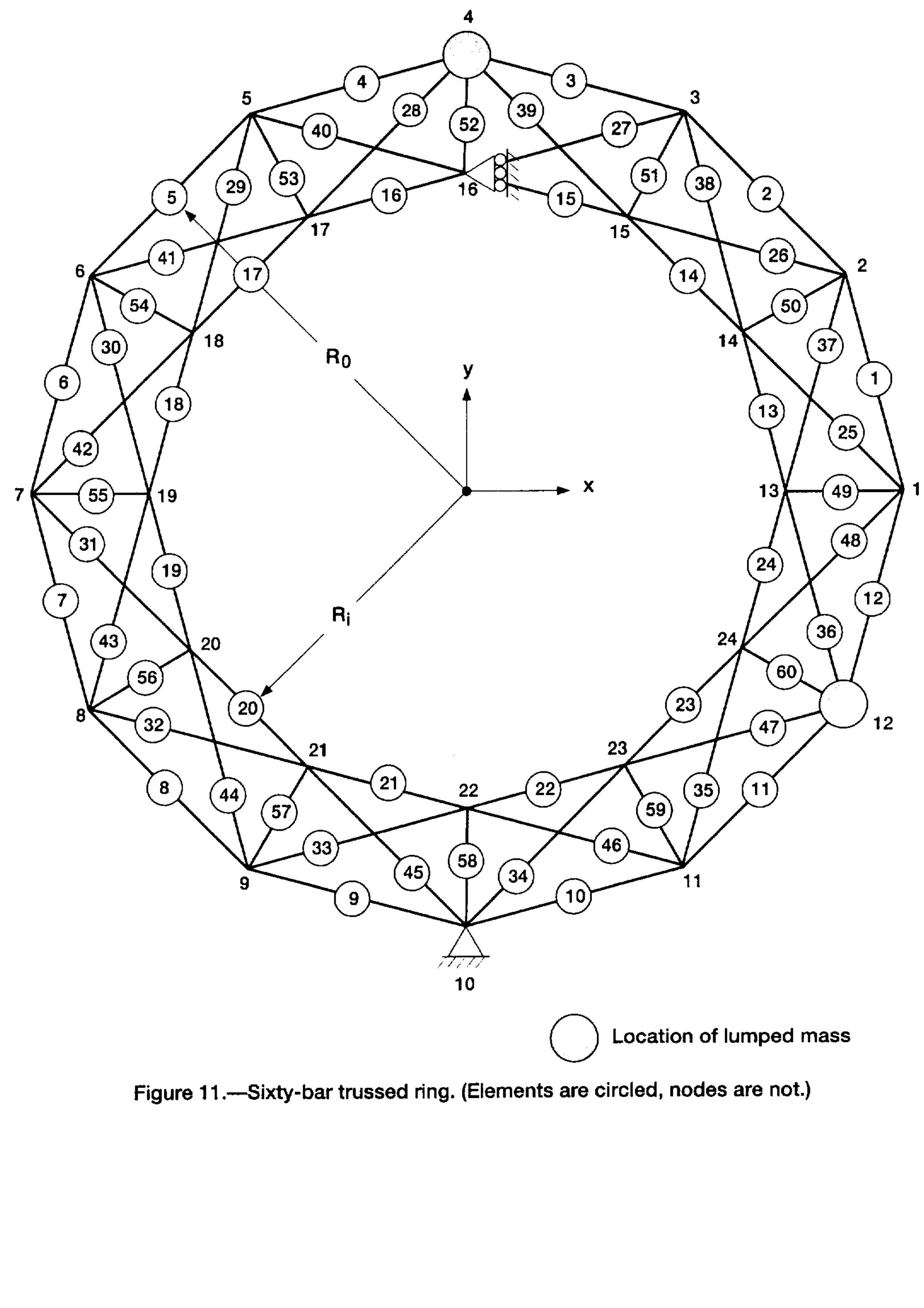

Figure 11.—Sixty-bar trussed ring. (Elements are circled, nodes are not.)

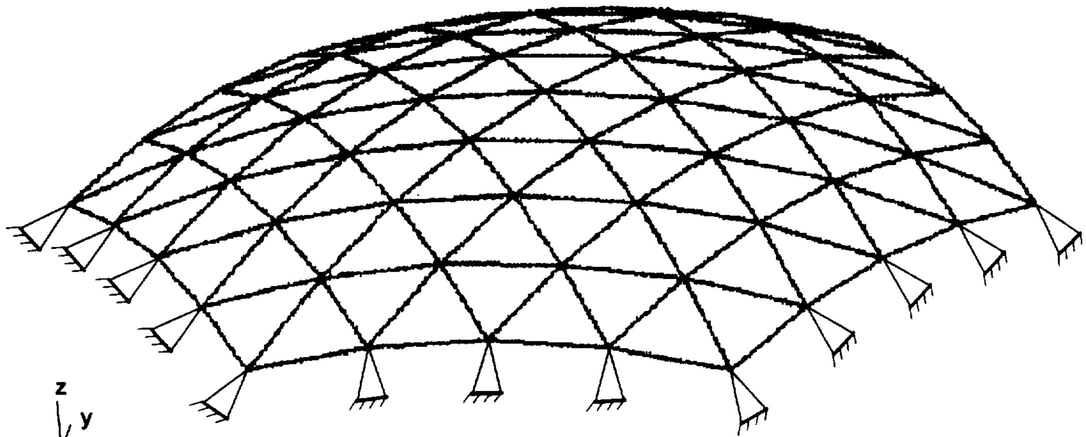

Figure 12.—Geodesic dome. (All boundary nodes are fully restrained. Supports are shown for three sides only. Supports for other sides are not shown.)

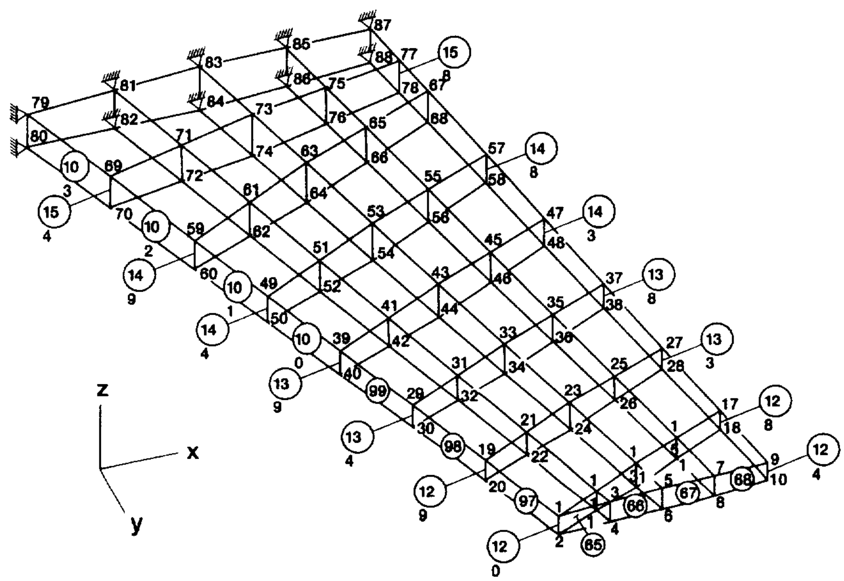

Figure 13.—Intermediate complexity wing. (Representative elements are circled, nodes are not.)

Examples P8a to P8d: Intermediate Complexity Wing

An intermediate complexity wing (refs. 3 and 18), shown in figure 13, was modeled with a total of 158 elements consisting of 39 bars, two triangular membranes, 62 quadrilateral membranes, and 55 shear panels. The wing is made of aluminum with modulus E = 10 500 ksi, and density $\rho = 0.1$ lb/in.^{3} The elements were grouped to obtain 57 linked design variables. The wing, which was subjected to two load conditions, had a total of 321 behavior constraints, (316 stress, four displacement, and one frequency). The optimum design for this problem was obtained from four different initial points, (1) initial design of unity; (2) initial design equal to the SUMT optimum design; (3) initial design equal to 150 percent of the SUMT optimum design; and (4) initial design which is infeasible at 50 percent lower than the SUMT optimum design. Results obtained for all four cases are summarized in table II (P8a to P8d). The optimum design was 387.76 lb and there were a total of 119 active constraints, (117 stress, one displacement, and one frequency). For initial design of unity (see table II, problem P8a), optimizers SUMT, SQP, IMSL, and NPSOL reached the optimum within a 3.7-percent error margin. Optimizer RG failed to solve the problem. Optimizer OC also failed to converge to the optimum (producing 20.1-percent over-design). Cray-YMP CPU time varied between 1075.21 sec for SQP to 8292.35 sec for NPSOL.

Discussion

For the purpose of this discussion, the 41 problems of the test bed are grouped as small, medium, and large. For the small problems (Group I), the number of linked design variables ranged between three and 19. Group I contains a total of 19 problems, which are designated as P1a to P5, P7, and P9 to P18. The normalized optimum weight for small problems obtained by each optimizer is depicted in figure 2. For medium problems (Group II), the number of linked design variables ranged between 20 and 39. There are 12 medium problems, which are designated as P6, P19 to P23, and P30 to P35. The normalized optimum weight for the medium problems obtained by each optimizer is illustrated in figure 3. Problems with more than 40 independent design variables are referred to as large problems (Group III). There are 10 large problems which are designated as P8a to P8d, and P24 to P29. The normalized optimum weight for large problems obtained by each optimizer is depicted in figure 4.

The following discussion is divided into five categories: (1) convergence to the optimum weight, (2) number of active constraints at optimum, (3) Cray-YMP8E/8128 CPU time required to solve the problem, (4) singularity in structural optimization, and (5) default optimization parameters.

Convergence to Optimum Weight

The normalized optimum weights for all 41 problems, obtained by the six optimizers are depicted in figures 2 to 4 for small, medium, and large problems, respectively. In these figures, unity represents optimum weight and more than unity indicates over-design, while less than unity is infeasible design.

For the purpose of comparison, a solution with constraint violation of less than one percent and weight which is within one percent of the best feasible design is considered optimum. A design is acceptable when the constraint violation is less than one percent and the weight is within five percent of the minimum obtained by the eight optimizers. Convergence to the optimum solution for each of the six optimizers follows.

(1) SUMT converged to optimum solution for 35 of 41 examples, which consisted of 17 small, nine medium, and nine large problems. SUMT failed for four problems. These are: one small problem (P5), two medium problems (P21 and P23), and one large problem (P8b). For both medium problems, the SUMT solution was more than one percent infeasible. For the large problem, SUMT gave an under-design of more than five percent.

(2) SQP of IDESIGN, successfully solved 32 of 41 examples, which consisted of 15 small, 10 medium, and seven large problems. This optimizer failed to give a feasible optimum design for three small problems (P15, P17, and P18), two medium problems (P19 and P22), and three large problems (P8c, P8d, and P25).

(3) IMSL optimizer DNCONG successfully solved 37 of 41 examples, which consisted of 17 small, 12 medium, and eight large problems. DNCONG of IMSL failed to optimize the intermediate complexity wing (problems P8c and P8d).

(4) NPSOL successfully solved 25 of 41 examples, which consisted of 13 small, eight medium, and four large problems. This optimizer failed (with an infeasible design over one percent) for: four small problems (P15 to P18); four medium problems (P23, P31, P32, and P34); and four large problems (P8d, P25, P26, and P28). It produced more than five percent over-design for large problem P8c.

(5) RG successfully solved 13 of 41 examples, which consisted of seven small, four medium, and two large problems. RG failed for 12 small problems. It also failed for seven medium problems and three large problems. The optimizer RG failed with well over 100 percent error in the optimum weight for 15 problems.

(6) OC successfully solved 16 of 41 examples, which consisted of six small, five medium, and five large problems. OC failed for nine small, two medium, and five large problems with an error in the optimum weight exceeding five percent, as well as for three medium problems with an infeasible design greater than one percent.

Number of Active Constraints at Optimum

The number of active constraints at the optimum for all examples is given in table I. Typically, different optimizers produced identical numbers of active frequency and active displacement constraints. However, the number of active stress constraints generated depended on the optimizer of choice. For example, with the geodesic dome problem (P7), the number of active stress constraints produced were 168 by SQP of IDESIGN, 162 by SUMT and NPSOL, and 156 by IMSL. Consider also the set of five examples depicted in table III that failed to converge, which produced minimum weights between 3.2 to 12.7 percent over- or under-designs. These examples produced correct numbers of displacement and frequency constraints, but failed to produce the correct numbers of active stress constraints. The deficiency in the number of active stress constraints ranged between three for problem P2 to 42 for problem P8a. For these problems the failure of the optimizers could be attributed to their inability to produce the correct number of active stress constraints. This aspect is also described in the section entitled, "Singularity in Structural Optimization" of this paper.

CPU Time Required For the Solution

The normalized CPU times on a Cray-YMP8E/8128 computer were recorded for a set of 14 examples. The normalization was with respect to SQP of IDESIGN except for problems P8c and P8d, which were normalized with respect to SUMT

TABLE III.—FOUR EXAMPLES THAT FAILED TO REACH OPTIMUM WEIGHT VERSUS BEST FEASIBLE DESIGN

Problem number	Optimization method	Percent over-design	Number of active constraints at optimum versus best feasible design		
			Frequency	Stress	Displacement
P2	OC	56	1	5	2
	vs. SQP	0	1	8	2
P6	RG	83.2	1	18	1
	vs. SQP	0	1	21	1
P8a	NPSOL	3.2	1	75	1
	vs. IMSL	0	1	117	1
P8c	IMSL	7.7	1	88	1
	vs. SUMT	0	1	109	1

TABLE IV.—PROBLEMS WITH ACTIVE CONSTRAINTS EXCEEDING THE NUMBER OF DESIGN VARIABLES

[Singularity can occur in each of these problems.]

Problem number	Description	Number of design variables	Number of active constraints at optimum
P2	Tapered ten-bar truss	10	11
P5	Antenna tower	6	20
P7	Geodesic dome	12	170
P8a	Intermediate complexity wing	57	119
P9	Ten-bar truss	10	11
P10	Ten-bar truss	5	9
P17	Cantilever membrane	16	19
P18	Cantilever membrane	16	35
P19	Sixty-bar trussed ring	25	38
P22	Sixty-bar trussed ring	25	30
P24	Stiffened ring	49	75
P27	Stiffened sixty-bar trussed ring	49	76
P30	Stiffened ring	24	28
P33	Stiffened ring	24	28

(table II and fig. 5). CPU time differed among optimizers. Even for a small problem (P1a), normalized CPU time differed from 1.0 for SQP to 22.069 for RG. For a medium problem (P6), normalized time differed between 1.0 for SQP to 3.899 for NPSOL. For a large problem (P8a), normalized CPU time varied from 1.695 for IMSL to 1.116 for SUMT and 1.000 for SQP of IDESIGN. We observed that variation in CPU time was rather mild for large problems.

Singularity in Structural Optimization

Singularity was identified for three situations (refs. 3 and 30):

(1) the number of active constraints exceed the number of design variables. Out of the 41 problems the 14 examples listed in table IV are prone to this type of singularity.

(2) linear functional dependencies among a small number of active stress constraints. This type of singularity is suspected to have occurred for some of the examples given in table III.

(3) linear functional dependencies among a small number of active stress and displacement constraints. The identification of this type of singularity by mere inspection may be difficult.

Singularity alleviation as discussed in references 8, 26, and 27 can reduce computation and improve reliability of optimizers (fig. 14).

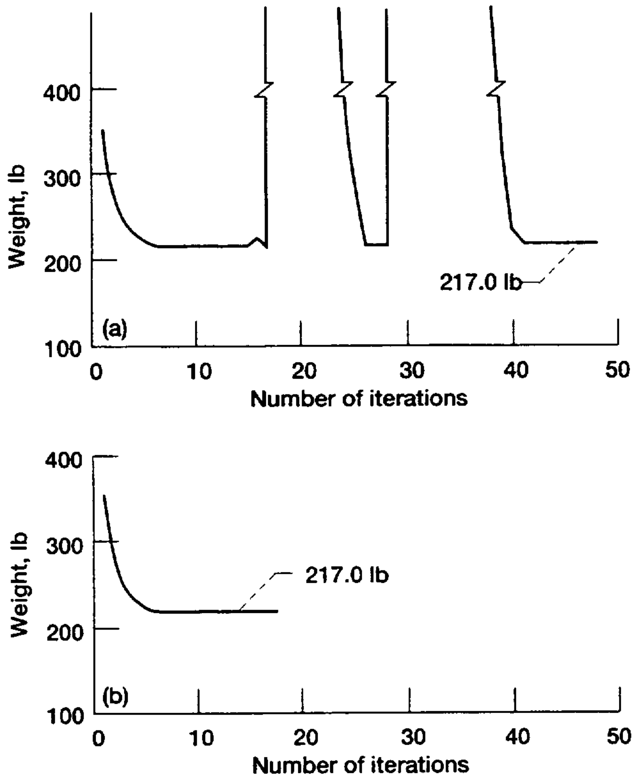

Figure 14.—Convergence characteristics of a three-bar truss, showing the merit function (weight) versus the number of iterations (breaks in the graph indicate weight up to the order of 10^5 lb). (a) Solution when singularity is disregarded. (b) Solution when singularity is alleviated.

Default Optimization Parameters

Default parameters (such as convergence criteria, step length, stop criteria, active constraint region, iteration limitations, etc.) specified by individual optimization codes, were used to solve the problems. When a problem failed, the default parameters were changed in an attempt to successfully solve the problem. In the solution of the 41 test bed problems, it was necessary to change the default optimization parameters quite often in order to reach the correct solution. On an overall basis, default parameters of SUMT, SQP, and IMSL algorithms were adequate for the solution of most problems. Most of the default parameters for RG and NPSOL were changed to improve their performances.

Concluding Remarks

None of the optimizers could successfully solve all the problems. Most optimizers, however, can solve at least one third of the examples. For large problems, the Cray-YMP CPU time was comparable among the optimizers that succeeded. Alleviation of singularity can improve the optimizer efficiency.

A single winner which can be called most reliable and efficient could not be identified. Overall, three optimizers (IMSL, SUMT, and SQP of IDESIGN) scored high marks. For small problems, four optimizers (IMSL, SUMT, SQP of IDESIGN, SLP, and NPSOL) satisfactorily solved more than fifty percent of the problems. For medium problems, four optimizers (IMSL, SQP of IDESIGN, SUMT, and NPSOL) produced correct solutions for at least half of the problems. For large problems three optimizers (IMSL, SUMT, and SQP of IDESIGN) were found to be reliable and efficient.

References

1. Guptill, J.D., et al.: CometBoards User's Manual, NASA TM–4537, 1996.
2. Vanderplaats, G.N.: DOT User's Manual, Version 2.04, VMA Engineering, 1989.
3. Patnaik, S.N.; Guptill, J.D.; and Berke, L.: Merits and Limitations of Optimality Criteria Method for Structural Optimization, NASA TP–3373, 1993.
4. Belegundu, A., et al.: Multiobjective Optimization of Laminated Ceramic Composites Using Genetic Algorithms. AIAA/USAF/NASA/ISSMO Proceedings of Symposium on Multidisciplinary Analysis and Optimization, 1994, pp. 1015–1022.
5. Lasdon, L.; and Waren, A.D.: GRG2 User's Guide, University of Texas at Auston, Texas, 1986.
6. IMSL User's Manual: MATH/LIBRARY: FORTRAN Subroutines for Mathematical Applications. Vol. 3, chapt. 8, 1987, p. 903.
7. Belegundu, A.D.; Berke, L.; and Patnaik, S.N.: An Optimization Algorithm Based on the Method of Feasible Directions. Structural Optimization J., vol. 9, 1995, pp. 83–88.
8. NAG FORTRAN Library Manual, MARK 15: E04UCF - NAG Fortran Library Routine Document, vol. 4, 1991.
9. Gabriele, G.A.; and Ragsdell, K.M.: OPT - A Nonlinear Programming Code in FORTRAN Implementing the Generalized Reduced Gradient Method, User's Manual. University of Missouri-Columbia, 1984.
10. Arora, J.S.: IDESIGN User's Manual, Version 3.5.2. Optimal Design Laboratory, University of Iowa, June 1989.
11. Miura, H.; and Schmit, Jr., L.A.: NEWSUMT: A FORTRAN Program for Inequality Constrained Function Minimization, User's Guide. (NASA Contract NGR-05-007-337.) NASA CR–159070, 1979.
12. Patnaik, et al.: A Cascade Optimization Strategy for Solution of Difficult Multidisciplinary Design Problems. AIAA–96–1628, 1996.
13 Thanedar, P.B., et al.: Performance of Some SQP Algorithms on Structural Design Problems-Sequential Quadratic Programming. Int. J. Numer. Meth. Eng., vol. 23, Dec. 1986, pp. 2187–2203.
14. Tseng, C.H.; and Arora, J.S.: On Implementation of Computational Algorithms for Optimal Design 1: Preliminary Investigation. Int. J. Numer. Meth. Eng., vol. 26, 1988, pp. 1365–1382.
15. Tseng, C.H.; and Arora, J.S.: On Implementation of Computational Algorithms for Optimal Design 2: Extensive Numerical Investigation. Int. J. Numer. Meth. Eng., vol. 26, 1988, pp. 1383–1402.
16. Schittkowski, K.: A Numerical Comparison of 13 Nonlinear Programming Codes With Randomly Generated Test Problems. Numer. Optimization Dynam. Syst., North-Holland Publishing Co., Amsterdam, 1980, pp. 213–234.
17. Hock, W.; and Schittkowski, K.: A Comparative Performance Evaluation of 27 Nonlinear Programming Codes. CO, vol. 30, 1983, pp. 335–358.

18. Canfield, R.A.; Grandhi, R.V.; and Venkayya, V.B.: Optimum Design of Structures With Multiple Constraints. AIAA J., vol. 26, Jan. 1988, pp. 78–85 .
19. Neill, D.J.; Johnson, E.H.; Herendeen, D.L.: Automated Structural Optimization System (ASTROS). AFWAL-TR-88-3028, Wright Patterson Air Force Base vol. II - User's Manual, April 1988.
20. Ragsdell, K.M.: A Survey of Some Useful Optimization Methods, Proceedings of the Design Engineering Technical Conference, A.H. Soni and H. Atmaran, eds., New York 1974, pp. 129–135.
21. Ragsdell, K.M.: The Evaluation of Optimization Software for Engineering Design. Lecture Notes in Economics and Mathematical Sys., Springer-Verlag, vol. 199, New York 1982.
22. Pshenichny, B.: Algorithms for the General Problem of Mathematical Programming, Kibernetica, no. 5, 1970, pp. 120–125.
23. Schittkowski, K.: NLPQL: A FORTRAN Subroutine Solving Constrained Nonlinear Programming Problems. Ann. Oper. Res., vol. 5, 1986, pp. 485–500.
24. Venkayya, V.B.; and Tischler, V.A.: ANALYZE: Analysis of Aerospace Structures With Membrane Elements. Tech. Rep. AFFDL-TR-78-170, 1978.
25. Patnaik, S.N.; and Gallagher, R.H.: Gradients of Behaviour Constraints and Reanalysis Via the Integrated Force Method. Int. J. Numer. Meth. Eng., vol. 23, 1986, pp. 2205–2212.
26. Patnaik, S.N.; and Srivastava, N.K.: On Automated Optimum Design of Trusses. Comp. Methods Appl. Mech. Eng., 1976, pp. 245–265.
27. Vanderplaats, G.N.; and Moses, F.: Automated Design of Trusses for Optimum Geometry. J. Struct. Div. Amer. Soc. Civil Eng. Proc., vol. 98, 1972, pp. 671–690.
28. Berke, L; and Khot, N.S.: Use of Optimality Criteria Methods for Large Scale Systems. Air Force Flight Dynamics Lab., Wright-Patterson Air Force Base, 1974.
29. Khot, N.S.: Optimization of Structures for Strength and Stability Requirements. Air Force Flight Dynamics Lab., Wright-Patterson Air Force Base, 1973.
30. Patnaik, S.N.; Guptill, J.D.; and Berke, L.: Singularity in Structural Optimization. Int. J. Num. Meth. Eng., vol. 36, 1993, pp. 931–944.
31. Gendy, A.S., et al.: Large Scale Structural Optimization with Substructuring in a Parallel Computational Environment. OAI/OSC/NASA Proceedings of Symposium on Application of Parallel and Distributed Computing, 1994, pp. 89–105.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
October 1996

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Performance Trend of Different Algorithms for Structural Design Optimization

5. FUNDING NUMBERS
WU–505–63–5B

6. AUTHOR(S)
Surya N. Patnaik, Rula M. Coroneos, James D. Guptill, and Dale A. Hopkins

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E–9690

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, D.C. 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM–4698

11. SUPPLEMENTARY NOTES
Surya N. Patnaik, Ohio Aerospace Institute, 22800 Cedar Point Road, Cleveland, Ohio 44142 and NASA Resident Research Associate at Lewis Research Center; Rula M. Coroneos, James D. Guptill, and Dale A. Hopkins, NASA Lewis Research Center. Responsible person, Surya N. Patnaik, organization code 5210, (216) 962–3135.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 39

This publication is available from the NASA Center for Aerospace Information, (301) 621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***
Nonlinear programming algorithms play an important role in structural design optimization. Fortunately, several algorithms with computer codes are available. At NASA Lewis Research Center, a project was initiated to assess performance of different optimizers through the development of a computer code CometBoards. This paper summarizes the conclusions of that research. CometBoards was employed to solve sets of small, medium and large structural problems, using different optimizers on a Cray-YMP8E/8128 computer. The reliability and efficiency of the optimizers were determined from the performance of these problems. For small problems, the performance of most of the optimizers could be considered adequate. For large problems however, three optimizers (two sequential quadratic programming routines, DNCONG of IMSL and SQP of IDESIGN, along with the sequential unconstrained minimizations technique SUMT) outperformed others. At optimum, most optimizers captured an identical number of active displacement and frequency constraints but the number of active stress constraints differed among the optimizers. This discrepancy can be attributed to singularity conditions in the optimization and the alleviation of this discrepancy can improve the efficiency of optimizers.

14. SUBJECT TERMS
Nonlinear programming algorithms; Structural design optimization; Sequential unconstrained minimization; Quadratic programming; Reduced gradient; Feasible directions; Singularity condition

15. NUMBER OF PAGES
19

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	